This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for monitoring a query during runtime, said query involving a plurality of join ~~operations;~~ operations, the method comprising the steps of:

running the query according to a first join order; and

concurrent with running the query, collecting performance statistics about each of the join operations.

2. (Original) The method according to claim 1, further comprising the step of:

changing the first join order, during running of the query, to a second join order based on the statistics.

3. (Original) The method according to claim 2, further comprising the step of:

collecting additional statistics about each of the join operations after the first join order is changed to the second join order.

4. (Original) The method according to claim 3, further comprising the step of:

changing the second join order to either the first join order or a third join order based on the additional statistics.

5. (Currently Amended) The method according to claim 1, wherein the plurality of join operations include:

a first join that includes a first table and a second table ~~join-from file and a first join-to file;~~

a second join that includes the first table and a third table ~~join-from file and a second join-to file~~.

6. (Original) The method according to claim 5, further comprising the steps of:

determining respective fan-in statistics for the first join and second join; and

changing the first join order to a second join order if the respective fan-in statistics indicate that the second join is more likely to cause fan-in than the first join.

7. (Original) The method according to claim 5, further comprising the step of:

determining respective fan-out statistics for the first join and the second join; and

changing the first join order to a second join order if the respective fan-out statistics indicate that the second join is less likely to cause fan-out than the first join.

8. (Original) The method according to claim 5, further comprising the steps of:

determining respective fan-in statistics for the first join and second join;

determining respective fan-out statistics for the first join and the second join; and

changing the first join order to a second join order based on a combination of the respective fan-in and fan-out statistics.

9. (Original) The method according to claim 1, comprising the steps of:

identifying a predetermined sample size;

performing the step of collecting statistics for the predetermined sample size;

evaluating the collected statistics; and

changing the first join order to a second join order based on the collected statistics.

10. (Original) The method according to claim 9, comprising the steps of:

    collecting additional statistics for substantially all of the query;

    comparing the additional statistics with the collected statistics; and

    adjusting the predetermined sample size, for use by a subsequent query,
according to results of the comparing step.

11. (Currently Amended) The method according to claim 1, further comprising
the steps of:

    running ~~an other~~ <u>another</u> query after the query; and

    selecting an initial join order for the other query based on the collected
performance statistics.

12. (Original) A method for optimizing a query join order during runtime, said
query involving a plurality of join operations, the method comprising the steps of:

    running the query according to a first join order; and

    concurrent with running the query, collecting statistics about each of the
join operations; and

    based on the collected statistics, selecting a preferred join order, while
running the query, such that the query continues to run according to the preferred
join order.

13. (Original) The method according to claim 12, further comprising the steps of:

    determining respective fan-in statistics for each of the join operations; and

    selecting the preferred join order based on the fan-in statistics.

14. (Original) The method according to claim 12, further comprising the steps of:

    determining respective fan-out statistics for each of the join operations;
and

    selecting the preferred join order based on the fan-out statistics.

15. (Original) The method according to claim 12, further comprising the step of:

performing the step of selecting a preferred join order after collecting statistics for a predetermined number of records from a table involved in the query.

16. (Original) An apparatus for executing a query comprising:

at least one processor;

a memory coupled with the at least one processor; and

a database engine residing in the memory and executed by the at least one processor, the database engine configured to run a query involving a plurality of join operations according to a first join order; and, concurrent with running the query, collect statistics about each of the join operations.

17. (Original) The apparatus according to claim 16, wherein the database engine is further configured to select a preferred join order based on the statistics.

18. (Original) The apparatus according to claim 17, wherein the database engine further comprises:

a manager configured to determine the select the preferred join order;

an execution engine, coupled with the manager, and configured to execute the query according to the preferred join order; and

a statistics collector, coupled with the manager and the execution engine, configured to monitor execution of each of the plurality of join operations, capture respective performance data; and communicate the respective performance data to the manager.

19. (Original) The apparatus according to claim 16, wherein the statistics include respective fan-in statistics and respective fan-out statistics for each of the join operations.

20. (Original) The apparatus according to claim 16, wherein the statistics are collected for a predetermined number of records from a table involved in the query.

21. (Currently Amended) A program product, comprising:
program code configured upon execution to perform the steps of:
    running the query according to a first join order;
    concurrent with running the query, collecting statistics about each of the join operations, and
    based on the collected statistics, selecting a preferred join order, while running the query, such that the query continues to run according to the preferred join order; and
a ~~tangible computer readable~~ signal bearing medium bearing the program code.

22. (Currently Amended) The program product according to claim 21, wherein the program code is further configured to:
    ~~determining~~ determine respective fan-in statistics for each of the join operations; and
    ~~selecting~~ select the preferred join order based on the fan-in statistics.

23. (Currently Amended) The program product according to claim 21, wherein the program code is further configured to:
    ~~determining~~ determine respective fan-out statistics for each of the join operations; and
    ~~selecting~~ select the preferred join order based on the fan-out statistics.